We claim:

1.  A system comprising:

    at least one thread;

    a pool of locks;

    at least one object, each object having an associated

variable; and,

    a recyclable locking mechanism to associate a lock with an

object using the associated variable of the object as a pointer,

upon a first request by a thread.

2.  The system of claim 1, wherein the recyclable locking

mechanism further is to deassociate the lock from the object upon

a second request by the thread.

3.  The system of claim 1, wherein the associated variable of

each of the at least one object comprises an integer.

4.  The system of claim 1, wherein the associated variable of

each of the at least one object comprises a set of high bits

defining the pointer to a lock and a set of low bits defining a

status variable.

5.  The system of claim 4, wherein the set of high bits

21

comprises 27 bits and the set of low bits comprises 5 bits.

6.    The system of claim 4, wherein the set of low bits is initially set to -1.

7.    The system of claim 4, wherein upon the first request the set of low bits is incremented by 1.

8.    The system of claim 7, wherein upon the set of low bits after incrementation by one being greater than 0, the associated variable has an in-use status by a thread such that the set of high bits points to a lock.

9.    The system of claim 7, wherein upon the associated variable after incrementation by one being less than 32, the associated variable has a spin status such that the set of high bits is currently in process of being set to a lock.

10.    The system of claim 4, wherein the recyclable locking mechanism further is to deassociate the lock from the object upon a second request by the thread,

such that upon the second request the set of low bits is decremented by 1.

11. A method comprising:

asserting an instruction by a thread to lock an object;

increasing an associated variable of the object, the associated variable having a set of high bits and a set of low bits; and,

determining whether the associated variable is greater than a boundary value.

12. The method of claim 11, further comprising initially setting the associated variable of the object to -1.

13. The method of claim 11, further comprising upon determining that the associated variable is less than the boundary value, waiting until the associated variable is greater than the boundary value.

14. The method of claim 11 further comprising upon determining that the associated variable is greater than the boundary value, using the set of high bits of the associated variable as a pointer to a lock for the object.

15. The method of claim 14, further comprising:

decrementing the associated variable of the object; and,

determining whether the associated variable is less than a

minimum threshold.

16. The method of claim 15, upon determining that the associated variable is less than the minimum threshold, recycling the lock.

17. A computer comprising:

a processor;

a computer-readable medium; and,

a recyclable locking mechanism program executed by the processor from the medium to associate a lock with an object using an associated variable of the object as a pointer.

18. The computer of claim 17, wherein the associated variable of the object comprises a set of high bits defining the pointer to a lock and a set of low bits defining a status variable.

19. A computer-readable medium having a recyclable locking mechanism program stored thereon for execution on a computer to associate a lock with an object using an associated variable of the object as a pointer.

20. The computer-readable medium of claim 19, wherein the associated variable of the object comprises a set of high bits defining the pointer to a lock and a set of low bits defining a

status variable.